

JURNAL PRAKTIKUM MODUL 8  
STUKTUR DATA 2022/2023 - 1  
"Circular Single Linked List"

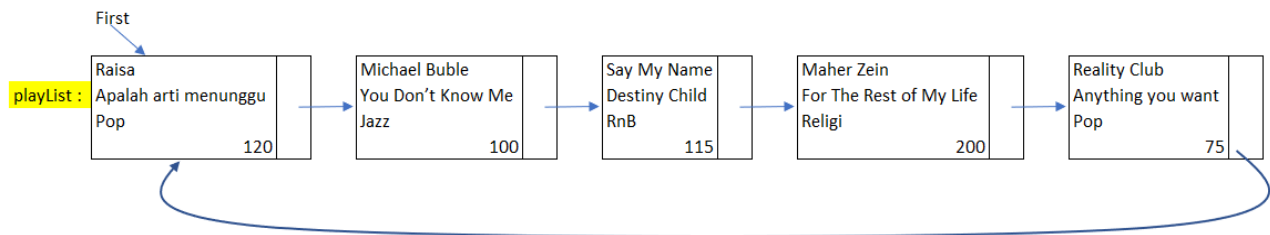
PENGUMUMAN :

PEKAN 9 (21-26 NOV 2022) AKAN DIADAKAN ASSESSMENT PRAKTIKUM MATERI MODUL 1-8  
LOKASI : LABORATORIUM PRAKTIKUM

INSTRUKSI PRAKTIKUM

1. Beberapa fungsi procedure telah ditanyakan dan dikerjakan saat Tugas Pendahuluan, jika masih ada yang belum berhasil silahkan coba kerjakan kembali saat praktikum dan sangat diharapkan meminta bantuan asisten jika ada yang kurang difahami. Silahkan menggunakan clue sintak pada soal ini atau menggunakan sintak yang sudah Anda buat di TP Anda masing-masing.
2. Jika seluruh fungsi di TP sudah berhasil, maka dapat lanjut ke fungsi procedure di sesi terbimbing

Sebuah Circular Linked List dengan pointer kepala First digunakan untuk menyimpan data playlist yang berisi data artis, judul, genre, dan jumlah pemutaran.



Buatlah ADT nya (CSLL.h, CSLL.cpp dan main.cpp dalam bahasa c++)!

1. CSLL.h (5 point)

```
Type infotype < artis: string
                    judul: string
                    genre: string
                    playtime: integer >
```

```
Type adr : pointer to element
Type element : < info: infotype
                next: adr >
Type listLagu : < first : adr >

[TP] Procedure create_List (in/out L: listLagu)
[TP] Procedure createElemen(in laguBaru: infotype; out pLagu:
adr)
[TP] Procedure insertFirst(in/out L: listLagu, in pLagu: adr)
[TP] Procedure insertLast(in/out L: listLagu, in pLagu: adr)
[TP] Procedure deleteFirst(in/out L: listLagu, out pLagu: adr)
[TP] Procedure deleteLast(in/out L: listLagu, out pLagu: adr)
[TP] Procedure showSemuaLagu(in L: listLagu)

//JURNAL
[JURNAL] Procedure tambahLagu(in/out L: listLagu, in pLagu: adr,
posisi: string)
[JURNAL-NIM GANJIL] Procedure showMostPlay(in L: listLagu)
[JURNAL-NIM GENAP] Function countGenre(L: listLagu, genre:
string) integer
[JURNAL] Function cariLagu (L: listLagu; artis, judul: string)
adr
[JURNAL-NIM GENAP] Procedure play(in L: listLagu, artisStart,
judulStart, artisEnd, judulEnd: string)
[JURNAL-NIM GANJIL] Procedure resetPlayList(in/out L: listLagu)
```

## 2. CSLI.cpp (60 point)

```
[TP] Procedure create_List (in/out L: listLagu)
{I.S. –
F.S. terbentuk list kosong}
Kamus Data
Algoritma
```

```
first(...) ← ...
```

**[TP] Procedure** createElemen(**in** laguBaru: infotype; **out** pLagu: adr)

(5 point)

{I.S. terdefinisi sebuah data lagu baru. Data jumlah playtime adalah 0  
F.S. element yang berisi data lagu baru alamatnya disimpan dalam pointer}

**Algoritma**

```
... ← new ...
info(...) ← ...
next (...) ← ...
```

**[TP] Procedure** insertFirst(**in/out** L: listLagu, **in** pLagu: adr)

(5 point)

{I.S. Terdefinisi sebuah list L yang bisa saja kosong, dan alamat element dari data lagu baru

F.S. lagu baru masuk kedalam list di posisi paling awal}

**Kamus Data**

X : ...

**Algoritma**

```
if (...!=NULL) then
    X ← first(L)
    While (next(X)!=...) do
        X ← next(X)
    endWhile
    next(X) ← ...
    ... ← first(L)
    first(L) ← ...
else
    first(L) ← ...
    next(...) ← first(L)
endIf
```

**[TP] Procedure** insertLast(**in/out** L: listLagu, **in** pLagu: adr)

(5 point)

{I.S. Terdefinisi sebuah list L yang bisa saja kosong, dan alamat element dari data lagu baru

F.S. lagu baru masuk kedalam list di posisi paling akhir}

**Algoritma**

```
if (...!=NULL) then
```

```

X ← first(L)
While (next(X) != ...) do
    X ← next(X)
endWhile
next(X) ← ...
... ← first(L)
else
    first(L) ← ...
    next(...) ← first(L)
endIf

```

[TP] **Procedure** deleteFirst(**in/out** L: listLagu, **out** pLagu: adr)

(5 point)

{I.S. Terdefinisi sebuah list L yang bisa saja kosong

F.S. data lagi yang berada di paling awal List dihapus dari list dan alamatnya disimpan ke dalam pointer}

**Kamus**

X : ...

**Algoritma**

```

if (first(L) == NULL) then
    Output("List Kosong")
else if (next(first(L)) == NULL) then
    // jika sisa satu element
    pLagu ← ...
    first(L) ← ...
else
    // jika list terdiri lebih dari satu element
    pLagu ← ...
    X = first(L)
    While (next(X) != first(L)) do
        X ← next(X)
    endWhile
    first(L) ← ...
    next(X) ← ...
    next(pLagu) ← ...
endIf

```

[TP] **Procedure** deleteLast(**in/out** L: listLagu, **out** pLagu: adr)

(5 point)

*{I.S. Terdefinisi sebuah list L yang bisa saja kosong*

*F.S. data lagi yang berada di paling akhir List dihapus dari list dan alamatnya disimpan ke dalam pointer}*

**Kamus**

X : ...

**Algoritma**

```

if (first(L) == NULL) then
    Output("List Kosong")
else if (next(first(L)) == NULL) then
    // jika sisa satu element
    pLagu ← ...
    first(L) ← ...

else
    // jika list terdiri lebih dari satu element
    X ← ...
    While (next(next(X)) != first(L)) do
        X ← ...
    endWhile
    pLagu ← ...
    next(X) ← ...
    ... ← ...

endIf

```

[TP] **Procedure** showSemuaLagu(**in** L: listLagu)

(5 point)

*{I.S. Terdefinisi sebuah list L yang bisa saja kosong*

*F.S. seluruh data artis, judul, genre, dan jumlah play time lagi tampil di layar}*

**Kamus Data**

P : ...

**Algoritma**

```

if (first(L) != NULL) then
    P ← first(L)
    Repeat
        Print(...)
        Print(...)
        Print(...)
        Print(...)
        P ← next(P)

```

```

        Until (...)
    else
        //Ketika list L kosong
        Print(...)
    endIf

```

**[JURNAL TERBIMBING] (5 point)**

**Procedure** tambahLagu(**in/out** L: listLagu, **in** pLagu: adr, posisi: string)

*{I.S. Terdefinisi list L yang merupakan list Lagu. List bisa kosong. Terdefinisi pula lagu yang akan dimasukkan ke List pada posisi awal/akhir}*

*F.S. Lagu baru masuk ke dalam list sesuai posisi yang diminta}*

**Kamus Data**

...

**Algoritma**

```

    if (posisi == "awal") then
        // panggil procedure insert first
        ...
    else
        // panggil procedure insert Last
        ...
    endIf

```

**[JURNAL TERBIMBING-NIM GANJIL] (10 point)**

**Procedure** showMostPlay(**in** L: listLagu)

*{I.S. terdefinisi sebuah List berisi lagu-lagu}*

*F.S. ditampilkan data nama artis dan judul lagu yang paling banyak diplay}*

*// clue : mencari play time terbanyak (Max)*

**Kamus Data**

P, Max: adr

**Algoritma**

```

    if (first(L) != NULL) then
        Max ← first(L)
        P ← next(first(L))
    Repeat

```

```

        If (info(P).playTime > ...) then
            Max ← ...
        endIf
        P ← next(P)
    Until (...)
        Print (info(Max).....)
else
        Print("List Kosong")
endIf

```

[JURNAL TERBIMBING-NIM GENAP] (10 point)

**Function** countGenre(L: listLagu, genre: string) integer  
{Function akan menghitung jumlah lagu pada genre tertentu}

#### Kamus Data

P: adr  
Count: integer

#### Algoritma

```

Count ← 0
if (first(L) != NULL) then
    P ← first(L)
    Repeat
        if (info(P).genre == ...) then
            Count ← ...
        endIf
        P ← Next(P)
    Until (P == first(L))
    return ...
endIf
return ...

```

[JURNAL TERBIMBING] (5 point)

**Function** cariLagu (L: listLagu; artis, judul: string) adr  
{Function akan mencari alamat dari lagu dengan artis dan judul yang diinginkan, jika tidak ditemukan return NULL}

#### Kamus Data

P: adr

#### Algoritma

```

If (first(L) != NULL) then
    P ← first(L)
    While (info(P)...!=...) AND (next(P) != first(L)) do
        P ← ...

```

```

        endwhile
        If (info(p)...==...) then
            return ...
        endif
    endif
    return NULL

```

#### [JURNAL TERBIMBING-NIM GENAP] (10 point)

**Procedure** play(**in** L: listLagu, artisStart, judulStart, artisEnd, judulEnd: string)  
*{I.S. terdefinisi sebuah List berisi lagu-lagu, alamat lagu awal, dan alamat lagi terakhir.  
 F.S. tampilkan ke layar nama artis dan judul lagu, mulai dari lagu pertama hingga lagu terakhir. Setiap kali artis dan judul ditampilkan maka jumlah play lagu tsb bertambah 1. Clue : gunakan function cariLagu  
 Contoh : lagu start di maher zein-for the rest of my life, end di Michael buble- You Don't Know Me, maka di layar akan muncul :*

*Maher Zein-For The Rest of My Life  
 Reality Club-Anything you want  
 Raisa-Apalah arti menunggu  
 Michael Buble-You Don't Know Me}*

#### **Kamus Data**

p, pStart, pEnd: adr

#### **Algoritma**

```

// cari alamat lagu awal dan lagu akhir
pStart ← cariLagu(L, artisStart, judulStart)
pEnd ← cariLagu(..., ..., ...)

//p pointer yang akan loop dari lagu awal sampai lagu akhir
p ← ...
Repeat
    Print(info(p).artis)
    Print(...)
    info(p).playtime ← ...+...
    p ← ...
Until (p == next(pEnd))

```

#### [JURNAL TERBIMBING-NIM GANJIL] (10 point)

**Procedure** resetPlayList(**in/out** L: listLagu)

*{I.S. Terdefinisi List Lagu yang tidak kosong.*

*F.S. Seluruh lagu dihapus satu persatu dari list L dan juga dari memory, list L menjadi kosong}*



**Kamus Data**

**Algoritma**

```
while (first(L) != NULL) do  
    // panggil procedure delete First atau Last  
    // hapus element dari memory  
endWhile
```

**3. Main.cpp (25 point)**

1. Buatlah sebuah list kosong
2. Isilah list tsb dengan N-data lagu yang Anda sukai. N sesuai dengan nomor urut kelas Anda + digit terakhir nim Anda. Misal Anda IF-05 dan digit terakhir Anda adalah 3 maka jumlah data pada list adalah 8. Setiap kali user memberikan data, minta juga mekanisme insert datanya (diakhir/diawal) kemudian **Panggil procedure tambahLagu**
3. Panggil procedure **show** agar terlihat hasil penambahan lagu Anda
4. **NIM GANJIL :**
  - Tampilkan nama artis beserta judul lagu yang paling banyak diputar
  - Reset playlist
5. **NIM GENAP :**
  - Minta ke user genre yang ingin dihitung jumlah kemunculannya di list lagu, lalu tampilkan hasil perhitungannya di layar
  - Minta user untuk menentukan range lagu yang akan diputar. Mintalah nama artis dan judul lagu pertama yang akan diplay, dan nama artis dan judul lagu terakhir yang akan diplay.
  - Minta berapa jumlah play list tsb diputar
  - Panggil procedure play